SQL

**DDL**-Data Definition Language

**DML**-Data Manipulation Language

**DCL**-Data Control Language

**DQL**-Data Query Language

**TCL**-Transcational control Language

**\c**- to connect database

**\l**-List of database

**\dt**- Show the database

***Create table:***

CREATE TABLE Detail\_pet (

Pet\_ID int PRIMARY KEY,

Name VARCHAR(20) NOT NULL,

Species VARCHAR(20)NOT NULL,

Breed VARCHAR(20) NOT NULL,

Health\_Status VARCHAR(20),

Status VARCHAR(20)

);

select \* from Detail\_pet;

INSERT INTO Detail\_pet

VALUES

(101,'Bella', 'Dog', 'Labrador','Healthy', 'Available'),

(102,'Whiskers', 'Cat', 'Persian', 'Healthy', 'Adopted'),

(103, 'Rocky', 'Dog', 'German Shepherd', 'Needs Vaccination', 'Available'),

(104, 'Coco', 'Rabbit', 'Holland Lop', 'Healthy', 'Available'),

(105, 'Bruno', 'Dog', 'Golden Retriever', 'Under Treatment', 'Available'),

('Kitty', 'Cat', 'Siamese', 'Healthy', 'Adopted');

**Alter Table:**

alter table Detail\_pet add column age int;

create table Adopt(

adopt\_id int PRIMARY KEY,

name VARCHAR(20) NOT NULL,

contract\_info bigint NOT NULL,

preferred\_pet\_type varchar(10) NOT NULL,

address varchar(20));

INSERT INTO Adopt VALUES

(1,'John Doe', 2334775533, 'Dog', '123 Main St, NY'),

(2,'Jane Smith', 2395710475, 'Cat', '456 Elm St, CA'),

(3,'Michael Brown', 9885382913, 'Rabbit', '789 Oak St, TX'),

(4,'Emily White', 9900556684, 'Dog', '321 Pine St, FL'),

(5, 'David Johnson', 4477558832, 'Cat', '654 Maple St, WA');

CREATE TABLE Adopt\_Record (

Adoption\_ID INT PRIMARY KEY,

Pet\_ID INT NOT NULL,

adopt\_id INT NOT NULL,

Adoption\_Date DATE NOT NULL,

FOREIGN KEY (Pet\_ID) REFERENCES Detail\_pet(Pet\_ID) ON DELETE CASCADE,

FOREIGN KEY (adopt\_id) REFERENCES Adopt(adopt\_id) ON DELETE CASCADE

);

**List all available pets sorted by species and then by name**

select Name from Detail\_pet where Status='Available' order by Species, Name;

**Find the total number of adoptions for each pet species, but only show species with more than one adoption**

select count(Status) from Detail\_pet where Status='Adopted' and count(Status) > 1 group by Species;

***write a sql query to retrieve the top three customer who are spending hightest total amount on orders your query should display the customer name, total amount spend and their ranking use apporiate sql function for aggregation and rankings***

**create customer table:**

CREATE TABLE Customers (

customer\_id INT PRIMARY KEY,

name VARCHAR(50) NOT NULL

);

**Create order table:**

CREATE TABLE Orders (

order\_id INT PRIMARY KEY,

customer\_id INT NOT NULL,

total\_amount DECIMAL(10,2) NOT NULL,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id) ON DELETE CASCADE

);

**Insert values:**

INSERT INTO Customers (customer\_id, name) VALUES

(1, 'John Doe'),

(2, 'Alice Smith'),

(3, 'Michael Brown'),

(4, 'Emily Davis'),

(5, 'David Wilson');

INSERT INTO Orders (order\_id, customer\_id, total\_amount) VALUES

(101, 1, 250.00),

(102, 2, 400.50),

(103, 3, 150.75),

(104, 1, 300.25),

(105, 4, 500.00),

(106, 5, 100.00),

(107, 3, 200.50),

(108, 2, 350.25),

(109, 4, 450.75),

(110, 1, 100.00);

select \* from Customers;

select \* from Orders;

**query for the question:**

SELECT

C.name AS customer\_name,

SUM(O.total\_amount) AS total\_spent,

RANK() OVER (ORDER BY SUM(O.total\_amount) DESC) AS ranking

FROM Customers C

JOIN Orders O ON C.customer\_id = O.customer\_id

GROUP BY C.customer\_id, C.name

ORDER BY rank;

DATE:21/03/25

**RANK()**

* Assigns a **ranking** to rows based on a specified order.
* **Skips ranks** if there are ties.
* Example: If two rows are ranked **1**, the next rank will be **3**.

**QUERY:**

# select \*,rank() over(order by salary) from emplo;

it skip the next number if the values are same it rank by the next's next number

**DENSE\_RANK()**

* Similar to RANK(), but **does not skip ranks** if there are ties.
* Example: If two rows are ranked **1**, the next rank will still be **2**

**QUERY:**

# select \*,dense\_rank() over(order by salary) from emplo;

it not give the gap between the same number and next number

**ROW\_NUMBER()**

* Assigns a **unique sequential number** to each row.
* **No gaps** in numbering, even if values are the same.

**QUERY:**

# select \*,row\_number() over(order by salary) from emplo;

it give a 1,2,3, order wise number for the values in a table

**PARTITION BY:**

* Divides the result set into **groups (partitions)** before applying a window function.
* Works like **GROUP BY**, but **does not collapse rows**.

**QUERY:**

- select \* , avg(salary) over(partition by dept) as dept\_avg\_salary, sum(salary) over (partition by dept) as dept\_sum\_salary from emplo;

**LAG()**

* Retrieves the **value from the previous row** within a partition.
* Useful for calculating **differences or trends over time**.

**QUERY:**

# select \*, lag(salary) over(partition by dept order by id) from emplo;

**LEAD()**

* Retrieves the **value from the next row** within a partition.
* Helps in **comparing current values with future values**.

**QUERY:**

# select \*, lead(salary) over(partition by dept order by id) from emplo;

**SUBQUERY:**

# select \*,(select avg(salary) from emplo e2 where e2.id=e1.id) from emplo e1;

**FIRST CHARACTER:**

#### **Using** SUBSTRING()

SELECT SUBSTRING(column\_name, 1, 1) AS first\_character FROM table\_name;

#### **Using** LEFT() **(MySQL, SQL Server)**

SELECT LEFT(column\_name, 1) AS first\_character FROM table\_name;

#### **Using** RIGHT() **for the Last Character**

SELECT RIGHT(column\_name, 1) AS last\_character FROM table\_name;

**SWAP:**

**Using select**

Select \* from table name,

Case

When column\_name= ‘y’ then ‘n’

When column name=’n’ then ‘y’

Else 0

End

**Whithout select:**

update Salary

set  sex= case

        when sex='m' then 'f'

        when  sex='f' then 'm'

END;

**SQL Questions & Answers**

### **1. How can I take the first character of a string using SQL?**

**Answer:** You can use the SUBSTRING() or LEFT() function.

SELECT SUBSTRING(column\_name, 1, 1) AS first\_character FROM table\_name;

Or in MySQL/SQL Server:

SELECT LEFT(column\_name, 1) AS first\_character FROM table\_name;

### **2. What does SUBSTRING(column\_name, 1, 1) do?**

**Answer:**

* SUBSTRING(column\_name, 1, 1) extracts a substring from column\_name.
* 1 → Start position (SQL indexes start from 1).
* 1 → Number of characters to extract.
* Example: If column\_name = 'Alice', it returns 'A'.

### **3. Is SUBSTRING a keyword in SQL?**

**Answer:** Yes, SUBSTRING is a built-in SQL function used to extract part of a string.

### **4. How do I extract a specific part of a string using SQL?**

**Answer:** You can use SUBSTRING() or MID() in MySQL.

SELECT SUBSTRING(column\_name, start\_position, length) FROM table\_name;

Example:

SELECT SUBSTRING('ChatGPT', 2, 3);

**Output:** hat

### **5. How do I get the last character of a string in SQL?**

**Answer:**

SELECT RIGHT(column\_name, 1) FROM table\_name;

Or using SUBSTRING():

SELECT SUBSTRING(column\_name, LENGTH(column\_name), 1) FROM table\_name;

### **6. How do I find the length of a string in SQL?**

**Answer:**

SELECT LENGTH(column\_name) FROM table\_name;

For SQL Server:

SELECT LEN(column\_name) FROM table\_name;

### **7. How do I concatenate two columns in SQL?**

**Answer:** Use CONCAT() function:

SELECT CONCAT(column1, column2) FROM table\_name;

For SQL Server:

SELECT column1 + column2 FROM table\_name;

### **8. How do I replace a specific character in a string using SQL?**

**Answer:** Use REPLACE() function:

SELECT REPLACE(column\_name, 'old\_value', 'new\_value') FROM table\_name;

### **9. How do I check if a column contains a specific substring?**

**Answer:**

SELECT \* FROM table\_name WHERE column\_name LIKE '%substring%';

### **10. How do I convert a column to uppercase or lowercase in SQL?**

**Answer:**

SELECT UPPER(column\_name) FROM table\_name; -- Converts to uppercase

SELECT LOWER(column\_name) FROM table\_name;

[**1211. Queries Quality and Percentage**](https://leetcode.com/problems/queries-quality-and-percentage/)

**ROUND:**

* ROUND() should be applied **outside** AVG() because AVG() aggregates before rounding.

### **Step-by-Step Breakdown:**

1. CASE WHEN rating < 3 THEN 1 ELSE 0 END
   * This checks if the rating is **less than 3**.
   * If **true**, it returns **1** (counting it as a poor query).
   * If **false**, it returns **0** (ignoring it in the count).
2. SUM(CASE WHEN rating < 3 THEN 1 ELSE 0 END)
   * This **adds up all the 1s**, giving the total number of poor queries.
3. NULLIF helps to avoid the zero because if zero in denominator it not works crt.

**Solution:**

 select query\_name, round(avg(rating /NULLIF(position,0)) ,2) as quality,

round((sum(case when rating < 3 then 1 else 0 end)/count(\*)) \* 100 , 2)  as poor\_query\_percentage from Queries group by query\_name;